## Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application.

## Listing of Claims:

1. (Currently Amended) A method for handling transaction messages in asynchronous data replication in a database system, the database system including a source node and a target node, ~~the~~ each transaction message having information concerning ~~at least one~~ a row change to a table copy at the source node, the method comprising:

~~(a)~~ determining ~~if the~~ whether a first transaction message has a dependency ~~any dependencies~~ on ~~at least one~~ a preceding non-completed transaction message, the first transaction message having a dependency on the preceding non-completed transaction when a row change associated with the preceding non-completed transaction requires application to a table copy at the target node prior to a row change associated with the first transaction message;

~~(b)~~ ~~if so,~~ responsive to the first transaction message having a dependency on the preceding non-completed transaction,

holding the first transaction message;

completing the preceding non-completed transaction message including applying the row change associated with the preceding non-completed transaction message to the table copy at the target node; and

~~(c) examining completed transaction messages to determine if the completed transaction messages remove the dependencies of the held transaction message; and~~

~~(d)~~ responsive to completing the preceding non-completed transaction message, releasing the ~~held~~ first transaction message ~~to be applied~~ and applying the row change associated with the first transaction message to ~~a~~ the table copy at the target node; and~~, if~~

~~the completed transaction messages remove the dependencies of the held transaction message.~~

responsive to the first transaction message not having a dependency on the preceding non-completed transaction, applying the row change associated with the first transaction message to the table copy at the target node without holding the first transaction message.

2. (Currently Amended) The method of claim 1, further comprising:

~~(e)~~ examining a plurality of transaction messages on a work queue by a plurality of agent threads;

~~(f)~~ applying in parallel row changes in each of the plurality of transaction messages by each of the plurality of agent threads;

~~(g)~~ updating a control table to indicate completion of the application of each of the plurality of transaction messages; and

~~(h)~~ placing each completed transaction message on a done queue.

3. (Currently Amended) The method of claim 2, further comprising:

~~(i)~~ examining each completed transaction message on the done queue;

~~(j)~~ determining if ~~the~~ completion of ~~the~~ a completed transaction message clears ~~the~~ dependencies of any ~~of the~~ held transaction messages dependent upon the completed transaction message; and

~~(k)~~ placing any of the held transaction messages onto the work queue, if the dependencies of the held transaction message have been cleared.

4. (Currently Amended) The method of claim 1, wherein ~~for each row change in the transaction message, the~~ determining ~~(a)~~ whether the first transaction message has a dependency

on the preceding non-completed transaction message comprises:

~~(a1)~~ determining that the row change in the first transaction message is an insert or a key update type of change;

~~(a2)~~ comparing a new replication key value in the row change in the first transaction message to an old replication key value of a row change in the preceding non-completed transaction message; and

~~(a3)~~ determining that the first transaction message has a dependency on the preceding non-completed transaction message ~~dependencies~~ if the new replication key value in the row change in the first transaction message is the same as the old replication key value in the row change in the preceding non-completed transaction message.

5. (Currently Amended) The method of claim 4, wherein ~~the~~ comparing ~~(a2)~~ a new replication key value in the row change in the first transaction message to an old replication key value of a row change in the preceding non-completed transaction message comprises:

~~(a2i)~~ comparing a hash value of the new replication key value in the row change in the first transaction message to a hash value of the old replication key value in the row change in the preceding non-completed transaction message.

6. (Currently Amended) The method of claim 1, wherein ~~for each row change in the transaction message, the~~ determining ~~(a)~~ whether the first transaction message has a dependency on the preceding non-completed transaction message comprises:

~~(a1)~~ determining that the row change in the first transaction message is a delete or a key update type of change;

~~(a2)~~ comparing an old replication key value in the row change in the first transaction message to a new replication key value in a row change in the preceding non-completed transaction message; and

~~(a3)~~ determining that the first transaction message has a dependency on the preceding non-completed transaction message ~~dependencies~~ if the old replication key value in the row change in the first transaction message is the same as the new replication key value in the row change in the preceding non-completed transaction message.

7. (Currently Amended) The method of claim 6, wherein ~~the~~ comparing ~~(a2)~~ an old replication key value in the row change in the first transaction message to a new replication key value in a row change in the preceding non-completed transaction message comprises:

~~(a2i)~~ comparing a hash value of the old replication key value in the row change in the first transaction message to a hash value of the new replication key value in the row change in the preceding non-completed transaction message.

8. (Currently Amended) The method of claim 1, wherein ~~for each row change in the transaction message, the~~ determining ~~(a)~~ whether the first transaction message has a dependency on the preceding non-completed transaction message comprises:

~~(a1)~~ determining that the row change in the first transaction message is an update type of change;

~~(a2)~~ comparing a new replication key value in the row change in the first transaction message to a new replication key value in a row change in the preceding non-completed transaction message; and

~~(a3)~~ determining that the first transaction message has a dependency on the preceding non-completed transaction message ~~dependencies~~ if the new replication key value in the row change in the first transaction message is the same as the new replication key value in the row change in the preceding non-completed transaction message.

9. (Currently Amended) The method of claim 8, wherein ~~the~~ comparing ~~(a2)~~ a new replication key value in the row change in the first transaction message to a new replication key value in a row change in the preceding non-completed transaction message comprises:

~~(a2i)~~ comparing a hash value of the new replication key value in the row change in the first transaction message to a hash value of the new replication key value in the row change in the preceding non-completed transaction message.

10. (Currently Amended) The method of claim 2, further comprising:

~~(i)~~ removing the completed transaction message from a receive queue.

11. (Currently Amended) The method of claim 10, wherein ~~the~~ removing ~~(i)~~ the completed transaction message from the receive queue comprises:

~~(i1)~~ deleting the completed transaction message from the receive queue as part of a two-phase commit synchronization with the application of the completed transaction message.

12. (Currently Amended) The method of claim 10, wherein ~~the~~ removing ~~(i)~~ the completed transaction message from the receive queue comprises:

~~(i1)~~ obtaining at least one entry in a control table at the target node indicating that the completed transaction message has been completed; and

~~(i2)~~ deleting the completed transaction message from the receive queue.

13. (Currently Amended) The method of claim 12, further comprising:

~~(i3)~~ removing the at least one entry from the control table.

14. (Currently Amended) A computer readable medium ~~with program instructions~~ containing a computer program for handling transaction messages in asynchronous data replication in a database system, the database system including a source node and a target node, ~~the~~ each transaction message having information concerning at least one row change to a table copy at the source node, the computer program comprising programming instructions for:

~~(a)~~ determining ~~if the~~ whether a first transaction message has a dependency ~~any dependencies~~ on ~~at least one~~ a preceding non-completed transaction message, the first transaction message having a dependency on the preceding non-completed transaction when a row change associated with the preceding non-completed transaction requires application to a table copy at the target node prior to a row change associated with the first transaction message;

~~(b)~~ ~~if so,~~ responsive to the first transaction message having a dependency on the preceding non-completed transaction,

holding the first transaction message;

completing the preceding non-completed transaction message including applying the row change associated with the preceding non-completed transaction message to the table copy at the target node; and

~~(c) examining completed transaction messages to determine if the completed transaction messages remove the dependencies of the held transaction message; and~~

~~(d)~~ responsive to completing the preceding non-completed transaction message, releasing the ~~held~~ first transaction message ~~to be applied~~ and applying the row change associated with the first transaction message to ~~a~~ the table copy at the target node; and~~, if~~

~~the completed transaction messages remove the dependencies of the held transaction message.~~

responsive to the first transaction message not having a dependency on the preceding non-completed transaction, applying the row change associated with the first transaction message to the table copy at the target node without holding the first transaction message.

15. (Currently Amended) The computer readable medium of claim 14, wherein the computer program further ~~comprising~~ comprises programming instructions for:

~~(e)~~ examining a plurality of transaction messages on a work queue by a plurality of agent threads;

~~(f)~~ applying in parallel row changes in each of the plurality of transaction messages by each of the plurality of agent threads;

~~(g)~~ updating a control table to indicate completion of the application of each of the plurality of transaction messages; and

~~(h)~~ placing each completed transaction message on a done queue.

16. (Currently Amended) The computer readable medium of claim 15, wherein the computer program further ~~comprising~~ comprises programming instructions for:

~~(i)~~ examining each completed transaction message on the done queue;

~~(j)~~ determining if ~~the~~ completion of ~~the~~ a completed transaction message clears ~~the~~ dependencies of any ~~of the~~ held transaction messages dependent upon the completed transaction message; and

~~(k)~~ placing any of the held transaction messages onto the work queue, if the dependencies of the held transaction message have been cleared.

17. (Currently Amended) The computer readable medium of claim 14, wherein ~~for each row change in the transaction message,~~ the programming instructions for determining ~~(a)~~ whether the first transaction message has a dependency on the preceding non-completed transaction message comprises programming instructions for:

~~(a1)~~ determining that the row change in the first transaction message is an insert or a key update type of change;

~~(a2)~~ comparing a new replication key value in the row change in the first transaction message to an old replication key value of a row change in the preceding non-completed transaction message; and

~~(a3)~~ determining that the first transaction message has a dependency on the preceding non-completed transaction message ~~dependencies~~ if the new replication key value in the row change in the first transaction message is the same as the old replication key value in the row change in the preceding non-completed transaction message.

18. (Currently Amended) The computer readable medium of claim 17, wherein the programming instructions for comparing ~~(a2)~~ a new replication key value in the row change in the first transaction message to an old replication key value of a row change in the preceding non-completed transaction message comprises programming instructions for:

comparing a hash value of the new replication key value in the row change in the first transaction message to a hash value of the old replication key value in the row change in the preceding non-completed transaction message.

19. (Currently Amended) The computer readable medium of claim 14, wherein ~~for each row change in the transaction message,~~ the programming instructions for determining ~~(a)~~ whether the

first transaction message has a dependency on the preceding non-completed transaction message comprises programming instructions for:

~~(a1)~~ determining that the row change in the first transaction message is a delete or a key update type of change;

~~(a2)~~ comparing an old replication key value in the row change in the first transaction message to a new replication key value in a row change in the preceding non-completed transaction message; and

~~(a3)~~ determining that the first transaction message has a dependency on the preceding non-completed transaction message ~~dependencies~~ if the old replication key value in the row change in the first transaction message is the same as the new replication key value in the row change in the preceding non-completed transaction message.

20. (Currently Amended) The computer readable medium of claim 19, wherein the programming instructions for comparing ~~(a2)~~ an old replication key value in the row change in the first transaction message to a new replication key value in a row change in the preceding non-completed transaction message comprises programming instructions for:

~~(a2i)~~ comparing a hash value of the old replication key value in the row change in the first transaction message to a hash value of the new replication key value in the row change in the preceding non-completed transaction message.

21. (Currently Amended) The computer readable medium of claim 14, wherein ~~for each row change in the transaction message,~~ the programming instructions for determining ~~(a)~~ whether the first transaction message has a dependency on the preceding non-completed transaction message comprises programming instructions for:

~~(a1)~~ determining that the row change in the first transaction message is an update type of change;

~~(a2)~~ comparing a new replication key value in the row change in the first transaction message to a new replication key value in a row change in the preceding non-completed transaction message; and

~~(a3)~~ determining that the first transaction message has a dependency on the preceding non-completed transaction message ~~dependencies~~ if the new replication key value in the row change in the first transaction message is the same as the new replication key value in the row change in the preceding non-completed transaction message.

22. (Currently Amended) The computer readable medium of claim 21, wherein the programming instructions for comparing ~~(a2)~~ comparing a new replication key value in the row change in the first transaction message to a new replication key value in a row change in the preceding non-completed transaction message comprises programming instructions for:

~~(a2i)~~ comparing a hash value of the new replication key value in the row change in the first transaction message to a hash value of the new replication key value in the row change in the preceding non-completed transaction message.

23. (Currently Amended) The computer readable medium of claim 15, wherein the computer program further ~~comprising~~ comprises programming instructions for:

~~(i)~~ removing the completed transaction message from a receive queue.

24. (Currently Amended) The computer readable medium of claim 23, wherein the programming instructions for removing ~~(i)~~ the completed transaction message from the receive

queue comprises programming instructions for:

~~(i1)~~ deleting the completed transaction message from the receive queue as part of a two-phase commit synchronization with the application of the completed transaction message.

25. (Currently Amended) The computer readable medium of claim 23, wherein the programming instructions for removing ~~(i)~~ the completed transaction message from the receive queue comprises programming instructions for:

~~(i1)~~ obtaining at least one entry in a control table at the target node indicating that the completed transaction message has been completed; and

~~(i2)~~ deleting the completed transaction message from the receive queue.

26. (Currently Amended) The computer readable medium of claim 25, wherein the computer program further ~~comprising~~ comprises programming instructions for:

~~(i3)~~ removing the at least one entry from the control table.

27. (Currently Amended) A system comprising:

a source node, wherein the source node sends a first transaction message concerning a committed transaction completed at a source table copy to a target node to asynchronously replicate the transaction; and

~~the target node,~~ wherein the target node comprises a receive queue, a browser thread, a work queue, a done queue, ~~at least one~~ an agent thread, and a target table copy,

wherein the first transaction message concerning the transaction is received on the receive queue,

wherein the browser thread examines the first transaction message on the receive queue to determine if the first transaction message has ~~any dependencies~~ a dependency on ~~at least one~~ a preceding ~~non-complete~~ non-completed transaction message, the first transaction message having a dependency on the preceding non-completed transaction when a row change associated with the preceding non-completed transaction requires application to a table copy at the target node prior to a row change associated with the first transaction message,

wherein the first transaction message is held by the browser thread ~~if~~ responsive to the first transaction message ~~has dependencies~~ having a dependency on the preceding non-completed transaction message,

wherein the preceding non-completed transaction is placed in the done queue when the row change associated with the preceding non-completed transaction message is applied to the table copy at the target node ~~done queue is examined for completed transaction messages to determine if the completed transaction messages remove the dependencies of the held transaction message~~,

wherein the ~~held~~ first transaction message is released and placed onto the done queue ~~if the completed transaction messages remove the dependencies of the held transaction message~~ responsive to the row change associated with the preceding non-completed transaction message being applied to the table copy at the target node,

wherein the first transaction message is not held by the browser thread responsive to the first transaction message not having a dependency on the preceding non-completed transaction message and the row change associated with the first transaction is applied to the table copy at the target node.

28-33. (Cancelled)